Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 1998 Proceedings

Americas Conference on Information Systems (AMCIS)

December 1998

A Case Study on the Software Culture at a Rank Startup: An Argument for Adopting a Mature, Corporate, and Process-Driven Mentality

Nguyen Ho
American University

Eugene McGuire
American University

A Case Study on the Software Culture at a Rank Startup: An Argument for Adopting a Mature, Corporate, and Process-Driven Mentality

Nguyen Ho
Eugene McGuire
American University

Abstract

Startup companies comprise a significant percentage of the U.S. software industry as the demand for software continues to exceed the supply available, minimizing the risk to new ventures. The challenges faced by startups are unique in each entrepreneurial enterprise. However, the implications for failing to manage the stages of software development are the same as those cited in mature organizations. The establishment and adherence to software processes can guide less experienced managers of startups and help these often struggling firms to avoid pitfalls that may reduce their chances of long-term survival and financial success. This conclusion is based upon examination of the software processes at IntraSoft Inc., a startup firm supplying intranet solutions.

Background

IntraSoft Inc., a Washington, DC based firm, was founded in November 1994 to capitalize on the explosive demand for Web consulting and development services. Co-founders Steven Brinker and Tyson Moore, recognizing that the phenomenal growth of the Internet was capturing the attention of every astute manager, started IntraSoft in the basement of Brinker's two-story townhouse.

Over the span of three years, IntraSoft grew from a small basement shop to a 30-person company. Housed in a 3,000 square feet office located four blocks from the White House, the Company's areas of specialization included: Web development, Internet connectivity, database applications, and multimedia design. These competencies strategically positioned IntraSoft to compete in the market to supply intranets—computer networks based on Internet technology.

Having developed intranets for several large customers, IntraSoft had achieved a suite of integrated software applications, which the Company intended to commercialize and market as a shrink-wrapped product. Called "KnowledgeShare", IntraSoft's flagship product allowed users to share files, schedule events, hold threaded discussions, post news items, and manage contacts using a Web browser as a standard client.

In May 1997, IntraSoft began struggling to reposition the Company from a consulting firm to a supplier of packaged commercial software. In order to focus more on software development, the Company retained more lucrative customer accounts, while referring smaller accounts to other firms. This decision caused a severe strain on IntraSoft's cash flow to the point where the Company was unable to meet payroll on a regular and timely basis.

Management

Upper managers included the Chairman, President & CEO, and Chief Operating Officer. Tyson Moore, Chairman, age 27, was the Company's primary source of financial investment. Prior to starting IntraSoft, Moore worked as a public policy and management consultant.

Steven Brinker, President & CEO, age 27, was the visionary leading IntraSoft. Some called him the "heart and soul of the Company." He was not a programmer, however, through personal interests and work experience he had become technically knowledgeable.

Matthew Jensen, Chief Operating Officer, age 36, handled the Company's legal affairs. His professional career as a lawyer included securities and commercial litigation as well as public policy consulting.

Middle managers at IntraSoft included the Controller, Director of Corporate Development, Director of Technology, Director of Consulting, and Director of Marketing. All middle managers had limited to no technology management experience and had been employed one year or less when the Company began the transition between business models.

Development Team

The development team consisted of one part-time and five full-time programmers. They were a young, talented, and easy-going group. Greg Wallace, Director of Technology, oversaw the development of KnowledgeShare. Prior to joining IntraSoft and being promoted to his title, Wallace worked as a systems and network administrator. He had no previous professional experience in software development.

Implications of Ad Hoc and Chaotic Software Processes

The software processes at IntraSoft were chaotic and ad hoc, as is the case with most startup companies. One cynical developer classified the team's processes as negative three on the SEI Capability Maturity Model. A more optimistic teammate said the Company was informally at level two. Such a large discrepancy of opinions was one indication of the Company's chaotic environment.

No formal procedure for requirements gathering was performed. Each developer was responsible for at least one of the six main modules in KnowledgeShare, and was given creative leeway in the design and development of the deliverable. Managers sometimes provided ideas for features and functionality, and the developers usually incorporated them. Customers were also an informal source of requirements when they provided feedback on their system.

The enforcement of coding standards contradicted the IntraSoft philosophy, "We hire smart people, and let them do what they do best." Other than the use of naming conventions, developers were free to program according to their own style. Some documented their logic with comments throughout the source code, while others chose not to document at all. This led to time lost when interns (some of whom were responsible for programming entire modules) left the Company. Often, code was rewritten when backtracking became more effort than could be reasonably justified.

The three most commonly cited ramifications of unmanaged software processes include failure to deliver on time, inability to meet customer expectations, and poor quality. These symptoms were all manifest at IntraSoft.

Product Delivery. In the beginning of Q4 1997 KnowledgeShare was still not ready to ship as a shrink-wrapped product. The software had been deployed only as a custom solution on several customer sites. Managers never set an exact date for when the product would ship commercially because their attitude was, "We'll deal with it when we have a contract on the table." To motivate employees, capture the interest of potential investors, and attract customer attention, managers often claimed greater progress towards the shipment date than had actually been achieved. In July when IntraSoft was making a marketing splash at Internet World '97, KnowledgeShare was months away from shipping. No documentation had been drafted, the feature set for the version that was supposedly shipping was not final, and a medium for delivering the software had yet to be discussed. Serious customers in the market for packaged software solutions would have been frustrated had they tried to place orders for timely delivery.

Customer Specifications and Requirements. A major drawback to the successful marketing of KnowledgeShare was the product specifications, which required Oracle technology for central data repository and Web request brokerage. KnowledgeShare's source code was written in PL/SQL, a language specific to communicating with Oracle databases. Although Oracle leads the relational database market, a significant number of organizations were customers of competing technologies, such as Informix, Sybase, and Microsoft SQL Server.

By default, IntraSoft began developing for Oracle databases because principal developers were certified in Oracle products. Lack of database independence caused many potential customers to dismiss KnowledgeShare on the basis of failure to comply with the specifications of their current system.

The informal process for gathering requirements did not provide significant insight into the features and functionality customers were seeking. This made it difficult for IntraSoft to position the Company strategically against competing firms, which had performed the necessary research to understand the requirements of the market.

Quality Assurance. Quality control was a continual challenge to IntraSoft's business success. Relationships with customers who had licensed KnowledgeShare were comparable to beta test sites. Customers reported bugs on a regular basis, and the fixes were used to improve the current version of the software.

Organizational Barriers to Software Process Improvement

In a textbook synopsis, initiating software process improvement requires strategic leadership, resource allocation, and team effort. Managers must possess the strategic foresight to recognize the importance and benefits of aligning sound software processes with desired business results. Once managers buy into the premise of establishing and adhering to software processes, they will budget and allocate the resources necessary to take action consistent with their beliefs. Through demonstrated commitment from managers, team members receive the support necessary to adopt common development procedures and tools to collaborate effectively on projects. Organizational barriers to software process improvement efforts arise when any of the previous components are missing.

Strategic leadership was not visibly evident in the daily management of IntraSoft. Upper managers, tended to discredit institutionalized business practices such as establishing guiding principles through written statements of vision, mission, and values. Without the establishment of basic foundations on which businesses depend, quick returns and immediate survival became the motivating force driving decision making, while quality was a secondary consideration.

The greatest impediment to adopting strategic leadership was the absence of experienced managers who examined the Company's strengths and weaknesses critically. The Company's early achievements, including the attainment of several large consulting contracts and media attention, bred an air of cockiness and a sense of predestined success among members of the management team.

Lack of resources was the second deficiency causing an organizational barrier to process improvement. As evidence of the severity of the situation, programmers developed on a single machine remotely through telnet. The development machine was

also responsible for hosting a couple of client Web sites and providing incoming-outgoing electronic mail service. Getting a basic sense of the performance of the software was difficult since the resources of the hardware were overburdened by tasks unrelated to the product itself. Lack of resources was also cited as a reason for not implementing methodological improvements, which do not require capital investment in new tools. Methodological improvements at IntraSoft should have included establishing coding conventions, documenting throughout the development process, specifying before coding, and testing modules thoroughly before integration.

Teamwork was the final deficient component hindering initiatives towards process improvement. Under tight office space constraints, most team members telecommuted from home and other locations. The Company's inability to house all employees in one central location had strong implications on the team culture. Communication and coordination between departments as well as individuals was a daily challenge. Constructive criticism, healthy banter, and creative brainstorming was often foregone due to the extra effort required to collaborate and share ideas in a normal work setting.

A Business Case for Software Process Improvement

Toward the close of Q3 1997, IntraSoft was ineffective in transitioning from a consulting firm to a packaged software development house. The reality of the situation forced managers to examine what they needed to do to guide the Company's efforts and bring KnowledgeShare to market according to the desired business model. Several business and software processes were put into place out of necessity and practicality.

Members of the development team were key in providing leadership and taking the initiative for improving the software processes, while managers focused on the Company's business processes. According to Jim Mills, lead developer, "The team did not implement process improvement sooner due to the Company's lack of human and financial resources. Putting software processes in place was inevitable. Processes were so chaotic that when customers called regarding software bugs, I didn't even know where to locate the other developers' source code if they were unavailable to make the necessary corrections."

The approach the development team adopted was to prioritize process improvement efforts based on necessity, practicality, and return on investment. Their primary source of guidance was a book purchased in the computer section at a local bookstore, titled *Rapid Development*. *Rapid Development* published by Microsoft Press provided over 100 practical methods for improving software processes. From those ideas, the team selected the ones they thought would provide the best return and implemented those processes. Although developers were familiar with process models such as the CMM and ISO 9001, they avoided them as a source of direction. "Process models tend to be too theoretical and rigid for our practical purposes and development environment," said Mills. "We take a more loose management approach."

Conclusion

Software is a unique commodity, which does not require a lot of resources to develop and use relative to most manufactured products. Although a poorly written software program may provide the same functionality as one that is well-written, the differences between the two deliverables become apparent beneath the surface. The well-written program may provide three levels of error checking and run at optimal efficiency, whereas this is never the case with a poorly written program. The enforcement of standards and adherence to process models will ensure high quality software. However, standards and processes must be established in the infancy stage of the development project. Unseasoned managers who have little experience organizing technical projects do not fully understand the implications of poor development practices. Recovering a software project once a large base of code has already been developed is extremely difficult, if not impossible.

Experienced management is a critical success factor in the development of new software firms. In light of the unique characteristics of software development, managers of such new enterprises must possess business savvy as well as the ability to lead and organize a large development project. If the latter skill is deficient, managers would be wise to recruit those with the ability to lead industrial size development projects.

Time is of the essence to new startup firms, which have a window of opportunity they must seize before it closes. Unlike established companies, startups do not have abundant resources to cushion their fall when projects fail. As a result, balancing the bureaucracy that the enforcement of standards and processes creates with the flexibility of more lenient management practices is important.

Allocating resources to ensure sound processes are incorporated into the development lifecycle is an investment in which firms can expect a return. A business must hold some competitive advantage relative to its competition, such as offering goods and services, which are better, cheaper, or delivered faster. Returns will be seen in costs saved from lower customer support and code rework. Products will also be easier to market on the basis of quality and timely delivery.